

ТЕНДЕНЦИИ ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ НА РЫНКЕ ЭЛЕКТРОННЫХ ТЕХНОЛОГИЙ

ХАХАНОВА И. В.

Предлагается обзор рыночных тенденций создания современных мобильных цифровых систем, рассматриваются эвристические законы, отражающие состояние развития микроэлектроники и компьютерной техники. Даются рекомендации для активизации научных исследований в наиболее перспективных направлениях, связанных с нуждами рынка электронных технологий.

1. Состояние рынка электронных технологий

В настоящее время продукция, поставляемая на рынок электронных технологий, используется во всех областях человеческой деятельности. Компании Intel, Motorola, Advanced Micro Devices, IBM, Sun Microsystems и Hewlett-Packard затрачивают миллионы долларов на разработку мощных процессоров, оперирующих уже 64- и 128-битовыми данными. Плотность размещения транзисторов в кристаллах микросхем достигла 200 млн на 1 см² [1]. У микропроцессоров, как и у других типов микросхем, наблюдается экспоненциальный рост возможностей, согласно закону Мура [2]. С такой же периодичностью упомянутые кристаллы увеличивают свою производительность на 100%. Это превратило разработку универсальных компьютеров в мощную лидирующую индустрию на рынке электронных технологий.

Однако ситуация в настоящее время изменяется в связи с ростом производительности средних и малых микропроцессоров. Очевидна рыночная тенденция – в течение ближайших 15 лет возможности, предлагаемые техническим прогрессом и рынком, превысят потребности пользователя. Поэтому корпорации, нацеленные только на выпуск мощных универсальных процессоров, придерживаясь закона Мура, рискуют потерять свою долю прибыли на рынке микроэлектроники. Чтобы этого не случилось, они уже сейчас реорганизуют часть своего производства на быстрый выпуск заказных (customized) микросхем.

Рынок микропроцессоров с годовым объемом продаж \$40 миллиардов имеет несколько уровней иерархии. Вершиной являются мощные микропроцессоры, используемые в серверах и рабочих станциях. Затем идут персональные компьютеры, где доминируют микропроцессоры фирмы Intel. Эти профессиональные чипы имели в США объем продаж \$23 миллиарда в 2001 году, после достигнутого максимума \$32 миллиарда в 2000 году, согласно информации от Semiconductor Industry Association (San Jose). Следующий уровень – микроконтроллеры дали производителям \$10 миллиардов прибыли в 2001 году. И последний – сигналъ-

ные процессоры (digital signal-processing), имеющие объем продаж \$4 миллиарда в 2001 году для использования в сотовых телефонах, DVD-плеерах.

Мощные процессоры применяются для трехмерных видеопроигрывателей, редактирования цифровых видеофайлов и speech-to-text трансляторов. Но в ближайшем будущем такие задачи будут решаться специализированными микропроцессорами. В большей части пользователи применяют компьютеры в качестве текстовых процессоров, для доступа к E-mail, Internet. Поэтому использование для этих целей высокопроизводительных процессоров Itanium и Pentium, Intel, Athlon, Advanced Micro Device, выполняющих один миллиард операций с плавающей запятой в секунду, в десятки и сотни раз избыточно. С другой стороны, надежность становится основным показателем конкурентоспособности, по которому выделяются фирмы IBM, Compaq и Hewlett-Packard, получающие основную прибыль на рынке электронных технологий. Вторым по значимости сейчас является показатель специализированной ориентации на решение конкретных задач – customization, предложенный компанией Dell. Когда возможности начинают превышать потребности данной области рынка, конкуренция смещает акценты с повышения общей производительности на усовершенствование свойств, за которые пользователь готов платить дополнительные деньги. Это – надежность, удобство и заказные свойства. Такие изменения переживают и другие отрасли промышленности, от автомобильной до универсальных ЭВМ.

Разработчики микропроцессоров предлагают свойства, превышающие потребности пользователей. Например, производители микросхем создают больше транзисторов на кристалле, чем это нужно для проектировщиков компьютеров. Еще в 1996 году компания National Technology Roadmap for Semiconductor обратила внимание на факт, что производство транзисторов ежегодно увеличивается на 60%, при этом их использование в новых проектах возрастает только на 20%. По этой причине и в области микроэлектроники происходит переориентация средств на возможности реализации только необходимых и заказных свойств.

Развивается модульность устройств, в которых сохраняется стандартизация интерфейсов между подсистемами. Такой подход уже реализован в нижних уровнях проектирования микропроцессора в виде системы на кристалле (System-on-chip или SoC), составленной из многократно используемых IP- (Intellectual Property)-модулей. Последние имеют различные размеры и функции, от простых, реализующих интерфейсы, до сложных 64-битовых микропроцессоров, имеющих несколько миллионов транзисторов. Компании (Tensilica, ARC Cores, Hewlett-Packard и STMicroelectronics) стали предлагать выбор компонентов и свойств в самих микропроцессорах. Указать необходимое множество команд можно через Web-сайт с помощью языка высокого уровня, такого как Verilog. Семей-

ство Hewlett-Packard HP/ST Lx позволяет определить число команд и операций, выполняемых параллельно, а также объединить Digital Signal Processing с другими полезными для человека функциями на одном кристалле.

Таким образом, прогресс в микроэлектронике диктует компаниям: 1) придерживаться стратегии модульности проектов, состоящих из многократно используемых и реконфигурируемых IP блоков; 2) предоставлять возможность реализации системы на одном кристалле, создавая заказные системы SoC; 3) сокращать энергоемкость, время проектирования и внедрения в производство микропроцессоров и систем SoC; 4) предоставлять широкие возможности для тестопригодного проектирования и верификации цифровых систем, существенно уменьшая время создания готового изделия.

2. Мобильное универсальное цифровое изделие

Уже порядка пяти лет существует устойчивая тенденция на рынке электроники к созданию портативных цифровых изделий, предназначенных для выполнения необходимых для человека функций. Они объединяются в понятие PDA (personal digital assistant) – персональный цифровой помощник: мобильный телефон, MP3-проигрыватель, просмотрщик электронной почты, цифровая камера, аудио-система, спутниковое радио, переносной телевизор, игровая приставка.

Возникает вопрос, можно ли создать универсальное цифровое изделие (universal digital assistant) с перечисленными выше функциями, которое автоматически настраивается на любой протокол передачи, приема и преобразования информации; самостоятельно выполняет upgrade конфигурации; длительный период времени работает автономно, на батарейках [3].

Ядром всех современных цифровых изделий является микропроцессор. За последние 25 лет он стал доступен не только компьютеру, но уже внедрился в стиральные машины, цифровые камеры, СВЧ-печи и тостеры. Однако любой самый мощный микропроцессор проигрывает по производительности или быстродействию специализированной микросхеме. Современные микропроцессоры могут выполнять миллиарды инструкций в секунду с частотой синхронизации 3000 МГц. Это в 600 раз выше быстродействия первых микропроцессоров в персональных компьютерах с потребляемой мощностью 80 Вт и более. При этом рассеиваемая мощность на 1 см² у последних была в пять раз выше, чем на поверхности кухонной электроплиты. Но все это не имело большого значения, пока компьютеры были стационарными и подключались к электрической сети. Высокий спрос на портативные компьютеры заставил производителей больше уделять внимания снижению потребляемой мощности микропроцессоров [4]. При этом акцент проектирования сместился с оценки производительности на единицу площади к производительности на Вт. Микропроцессор, с точки зрения потребляемой мощности, стал неэффективным.

Поэтому он утратил позицию устройства, выполняющего все основные функции, и перешел в разряд управляющего ядра.

Исходя из предложения рынка микроэлектроники наиболее эффективными с позиции быстродействия являются: ASIC – Application-Specific Integrated Circuit – специализированные микросхемы, разработанные одной фирмой для конкретного изделия; ASSP – Application Specific Standard Product – специализированное устройство, которое может быть использовано многими разработчиками. Например, Analog Devices, Infineon, Motorola, Qualcomm и Texas Instrument выпускают микросхемы ASSP, которые любой производитель может использовать для создания, например, сотовых телефонов. Такие микросхемы могут быть в тысячи раз эффективней по стоимости и производительности на ватт, чем микропроцессоры. Однако несмотря на упомянутые преимущества, специализированные микросхемы не обладают гибкостью реконфигурирования. Вследствие высокой стоимости масок кристалла разработка и выпуск новой микросхемы стоит около 2 миллионов долларов.

Своего рода компромиссом являются программируемые логические интегральные схемы (ПЛИС или PLD), обладающие производительностью специализированных микросхем и гибкостью микропроцессоров. Такие изделия состоят из двух слоев: первый содержит конфигурируемые для конкретной функции линии и элементы, второй – память (ре-)конфигурации или структуры цифрового изделия.

Если рассматривать только перепрограммируемые микросхемы, использующие в качестве памяти реконфигурации SRAM, то здесь структура функций проекта загружается однажды в момент инициализации, после включения питания. Отдельные чипы предоставляют возможность для неполной реконфигурации, когда часть элементов и связей может быть изменена, в то время как остальные останутся неизменными. Интерфейсная микросхема должна иметь возможность адаптации к различным протоколам без влияния на остальные ее функции. Интересны для разработчиков цифровых устройств и динамически реконфигурируемые чипы [5-7], когда любые связи и элементы могут изменяться во время работы микросхемы. Но и такие кристаллы уже не подходят для реализации универсального цифрового устройства. Они должны повысить свое быстродействие и снизить энергопотребление. При этом за свою гибкость реконфигурации ПЛИС расплачиваются некоторым уменьшением производительности: 20 транзисторов программируемого кристалла приходится на один транзистор ASIC. Память реконфигурации может занимать до 70% общего числа транзисторов микросхемы. Длинные линии связи вентиляей и транзисторные программируемые коммутационные матрицы значительно замедляют скорость выполнения операций в PLD. На практике сложное цифровое устройство, реализованное на современных ПЛИС, представляет собой изделие, занима-

юшее большую площадь на кристалле, с низким быстродействием, высокой потребляемой мощностью и к тому же дорогостоящее (более \$1000), что делает PLD непригодными для создания портативных компьютеров.

Компании Altera и Xilinx, занимаясь устранением обозначенных недостатков, расширили возможности PLD в целях обеспечения их конкурентоспособности с микропроцессорами и сигнальными процессорами. Последние разработки ПЛИС с высокоскоростными вход-выходами, блоками памяти, умножителями и встроенными микропроцессорами представляют серьезную альтернативу мощным стационарным изделиям. К сожалению, эти дополнительные свойства существенно увеличивают потребляемую мощность и делают их непригодными для создания портативных устройств. Тем не менее, только PLD в ближайшем будущем предоставят возможности для разработки универсального цифрового устройства, если в ближайшие 2-3 года удастся сократить избыточность входов и соединений, ускорить и упростить процесс полной или частичной реконфигурации. В этом случае функции управления остаются резидентными, остальные загружаются по мере надобности. Следует также заметить, что компании-производители микросхем стараются сохранить статус-кво для пользователей, поэтому они не стремятся кардинально менять архитектуру чипов. Процесс продвижения на рынок новой микросхемы сложен, связан с определенным финансовым риском и требует больших материальных затрат, что могут себе позволить только крупные компании, такие как Intel, Motorola, NEC, Nokia и Texas Instruments, занимающиеся разработками в области реконфигурируемых процессоров и цифровых систем.

Интересное направление повышения эффективности используемой площади кристалла связано с динамической реконфигурируемостью функций. В памяти находятся описания нескольких полезных функций, каждая из которых может быть сконфигурирована на кристалле с помощью супервизора за один цикл. При этом осуществляется экономия энергопотребления, благодаря выполнению в конкретный момент только одной функции. Кроме того, частичная (полная) динамическая реконфигурируемость позволяет повысить эффективность использования площади кристалла в предположении, что все функции будут востребованы на длительном временном интервале. Компании-разработчики (start-up companies), такие как QuickSilver, Elm Technology, IBM, Matrix Semiconductor, Tezzaron Semiconductor, Ziptronic, работают над созданием многослойной микросхемы для портативных устройств, в которой внутренние соединения выполняются тысячами вертикальных линий. Это позволит значительно снизить потребляемую мощность и повысить быстродействие выполнения функций.

Другой путь усовершенствования ПЛИС — замена исчерпавшей себя SRAM-памяти, имеющей высокую потребляемую мощность, на более эффектив-

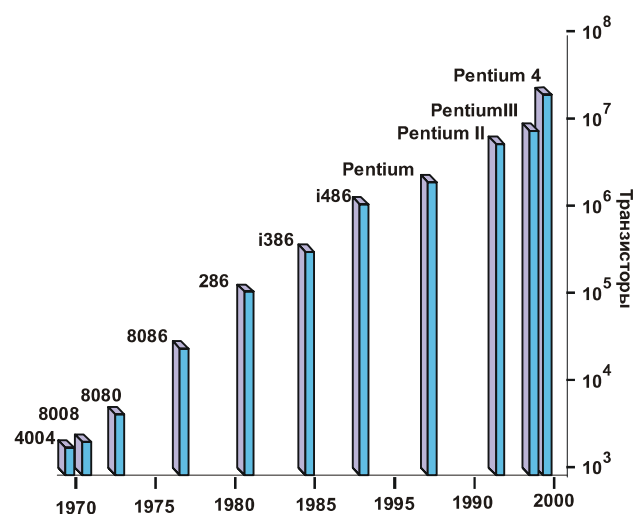
ную, энергонезависимую и быстродействующую [8]. Наиболее перспективными являются: магниторезистивная память, ферроэлектрическая и память на аморфных полупроводниках. Использование такой памяти позволит также улучшить и функции защиты информации при отключении энергопитания. Таким образом, новый тип PLD позволит решить задачу создания портативного универсального цифрового устройства. При этом основной функцией микропроцессора будет управление [9].

3. Законы развития рынка электронных технологий

За время, прошедшее между открытием электрона и появлением микросхем, содержащих более 200 миллионов транзисторов на кристалле, было сделано множество математических и эвристических наблюдений, содержательных высказываний и устойчивых прогнозов. Однако 90 % всех утверждений — констатирует закон писателя научной фантастики Теодора Стежна (Sturgeon's Law) — является полной чепухой. Самым известным из них является закон, сформулированный военно-космическим инженером Эдвардом Мерфи (Edward A. Murphy Jr.): "Если существуют две или несколько возможностей, и одна из них может привести к катастрофе, то кто-нибудь ее выберет." Полвека активного использования полупроводниковых приборов привело к созданию эмпирических законов, пять из которых — востребованы на рынке [10].

3.1. Закон Мура (Moore's Law)

"Число транзисторов в микросхемах ежегодно удваивается". Закон Мура был представлен в документах компании Intel 38 лет назад Гордоном Муром (Gordon E. Moore). Еще в 1965 году он отметил, что нет технических препятствий для усовершенствования существующих технологий (фотолитографии) и предсказал, что через 10 лет число транзисторов в микросхемах достигнет 65000.



Однако к концу 1975 года микросхемы имели едва ли десятую часть предсказанного объема. Период времени для удвоения числа транзисторов на самом деле составил 17 месяцев. К 1985 году данный интервал достиг 22 месяца, а к 1995 году — 32

месяца. Затем наметилось сокращение периода до 22-24 месяцев, который действителен и в настоящее время (G. Dan Hutcheson, CEO of VLSI Research). Специалисты компании Intel оценивают настоящий период времени удвоения числа транзисторов в микросхеме, равный 18 месяцам. Экономическая основа закона Мура: стоимость компонента для простых схем обратно пропорциональна их числу. Но уменьшение стоимости прекращается при увеличении сложности схем. Таким образом, может наступить момент, когда число транзисторов в схеме наращивать экономически нецелесообразно. Поэтому если закон Мура перестанет действовать, то это произойдет по экономическим, а не по техническим причинам. Мур утверждает, что экспоненциальная тенденция роста производительности и числа транзисторов в микросхеме не может существовать постоянно, но ее изменение может быть отсрочено. По пессимистическим прогнозам полупроводники будут преобладать по крайней мере ближайшие 15 лет. Затем ожидается переход к новой технологии, которая может не иметь обозначенных экономических свойств (полупроводников), однако это произойдет не скоро. Использование закона Мура также ставит вопрос: не изменяется ли его действие от описания к предсказанию? Закон просто описывает реальность или влияет на будущее? Так или иначе, но электронная промышленность использует закон для составления планов развития (roadmap) и прогнозирования конструктивных особенностей чипов: минимальная ширина линий, максимальный размер пластины (wafer), точность инструментов, чистота производственных помещений. Производители полупроводниковых устройств и в дальнейшем будут увеличивать число транзисторов на кристалле до тех пор, пока это экономически целесообразно и позволительно для существующих средств производства микросхем.

3.2. Закон Рока (Rock's Law)

“Стоимость оборудования для выпуска полупроводниковых кристаллов удваивается каждые 4 года”. Иногда его называют вторым законом Мура, однако сам он приписывает авторство Артуру Року (Arthur Rock), инвестору Intel. Тем не менее на практике оказалось, что стоимость оборудования остается постоянной, а возрастает его производительность в процессе модернизации, измеряемая числом выпущенных в час пластин (wafer) (20 штук – 1990 годы, 40-50 – наши дни). Благодаря возросшей производительности без увеличения стоимости оборудования цена компонента на кристалле изменилась от 1 цента за один транзистор до 1 доллара за миллион. Как это влияет на стоимость персонального компьютера?

3.3. Закон Макроуна (Machrone's Law)

“PC, который вы хотите купить, всегда стоит \$5000”. Сформулирован в 1984 году Биллом Макроуном (Bill Machrone), обозревателем журнала PC Magazine. Построив график стоимости компьютеров с 70 по 90 годы, он обратил внимание, что их

цена в указанном временном интервале остается постоянной и равной \$ 5000. В настоящее время эта цифра уменьшилась до \$ 3000. Сейчас приемлемый для работы компьютер стоит порядка \$ 1000, но с полной конфигурацией и периферией его стоимость составляет \$ 2000-3000. Уменьшение стоимости компьютеров привело к значительному увеличению числа пользователей PC. При этом каждая новая машина подключается к локальной сети или к Internet. Какие преимущества дает сеть для пользователя?

3.4. Закон Меткафа (Metcalf's Law)

“Рост значимости сети определяется квадратом числа её пользователей”. Сформулирован Робертом Меткафом (Robert Metcalfe), изобретателем стандарта Ethernet и основателем компании 3Com Corp. в 1980 году. В отличие от предыдущих, этот закон не может быть определен количественно, поскольку значимость или ценность информации трудно оценить числом, поскольку она различна для конкретных пользователей. Нежелательные для вас клиенты-contaminant понижают значимость сети. Такими могут быть маркетинговые фирмы, распространяющие ненужную для вас информацию, spam e-mail или неисправные узлы, нарушающие маршрутизацию и приводящие к потере передаваемых данных. В настоящее время отмечены 4 проблемы сетей: 1) насыщенность (saturation) – новые ее клиенты не могут принести в неё новую информацию; 2) какофония (cacophony) – непредсказуемость взаимодействия между группами клиентов в процессе телеконференций, дискуссий, порождающих сотни сообщений; 3) кластеризация (clustering) – разбиение клиентов на группы, каждая из которых использует только часть сети; 4) поиск (search) – становится все более времязатратным и дорогостоящим, поэтому большая часть информационных богатств сети остается недоступной на практике для пользователя. Существуют некоторые пути частичного решения данных проблем: 1) подключать к сети абонентов, обладающих ценной информацией; разделять сеть на фрагменты, например, по тематике передаваемой информации; 3) исключать из сети контаменанты (contaminant), засоряющие сеть, или фильтровать генерируемую ими информацию.

3.5. Закон Вирта (Wirth's Law)

“Снижение скорости программного обеспечения происходит быстрее роста скорости аппаратуры” (Software is slowing faster than hardware is accelerating). Сформулирован в 1995 году Никлаусом Виртом (Niklaus Wirth), автором языка программирования Паскаль. Эта же идея ранее звучала в анонимной шутке на манер библейского стиха: “Groves giveth, and Gates taketh away – что сад принес, рот забрал”. Благодаря Энди Гроуву (Andy Grove) скорость аппаратуры возрастает, а Билл Гейтс (Bill Gates) увеличивает сложность приложений, которая замедляет их быстроедействие. Текстовые редакторы 70-х годов требовали 8000 байт ОЗУ, а для современных аналогов необходимый объем памяти уве-

личился в 100 раз. Эти программы нельзя было бы использовать при отсутствии повышения быстродействия “железа” в тысячи раз. Следует помнить, что большинство функций программных приложений являются избыточными и никогда не будут востребованы пользователем. Но почему он это терпит? Существуют 2 причины: 1) удивление невежественного пользователя вычислительными способностями программы, который не в состоянии отличить полезные функции от бесполезных; 2) заинтересованность компаний, создающих программное обеспечение. Полезные свойства, такие как цвет, выпадающие иконки делают программы дружелюбными, поэтому они не нуждаются в сложной документации. Даже непрофессионал может научиться работать с такой программой, но сделав это он попадает на потребительский крючок на долгое время. Кроме того, требование к более быстрому выпуску программы на рынок не дает программисту времени на ее тщательную доработку, в том числе и в целях уменьшения ее размера.

Конечно, эвристические законы и правила не ограничиваются пятью упомянутыми. Еще один, касающийся пропускной способности Internet — закон Якоба Нильсена (Nilsen’s Law) — был опубликован в 1998 году. Он гласит, что доступная для профессиональных пользователей пропускная способность сети будет возрастать на 50% ежегодно, но разработчики Web-сайтов не получают этого преимущества, которое будет компенсироваться увеличением числа клиентов и замедлением программного обеспечения.

Подтверждением жизненности приведенных законов может служить исследование, проведенное доктором Е. Зорианом (Yervant Zorian, Virage Logic). Занимаясь анализом рынка и производства цифровых и микропроцессорных систем, он приходит к следующим обобщениям [11]:

- 1) Количество транзисторов на кристалле возрастает быстрее, чем их необходимый объем для проектирования сложных цифровых систем.
- 2) Инвестиции в технологический процесс преобладают над капиталовложениями в создание новых цифровых и микропроцессорных систем.
- 3) Материальные затраты на разработку программного обеспечения встроенной системы (embedded system) составляют 80% от ее стоимости.
- 4) Затраты на тестирование имеют экспоненциальную зависимость от объема (стоимости) выпуска продукции.
- 5) Число инженеров-верификаторов (тестеров) не менее, чем в 2 раза превышает количество проектировщиков цифровых и микропроцессорных систем.

Каждый тезис имеет методологическое значение для перспектив развития радиоэлектронного производства и рынка развивающихся стран, поскольку через два-три года их компании с необходимостью должны учитывать данные тенденции в планировании своей деятельности.

4. Gecko — пример создания гибких реконфигурируемых систем на кристалле

Межуниверситетский центр микроэлектроники (Interuniversity MicroElectronic Center, Leuven, Belgium) разработал мультимедийную платформу (Gecko), которая может быть реконфигурирована на PDA в MP3-плеер, игровую приставку или мобильный телефон [12,13]. Разработка выполнена благодаря поддержке четырех коммерческих партнеров, среди которых Xilinx Inc, крупнейшая компания по выпуску FPGA, и фирма, выпускающая кристаллы Infineon Technologies AG (Munich). Проект разрабатывался в течение трех лет. За последующие пять лет планируется разместить проект в одной микросхеме и довести его до конечного пользователя. Устройство Gecko состоит из трех частей. Первая — блок FPGA, предоставляющий максимальные возможности для реализации задач. Вторая часть — блок специализированных микросхем ASIC, отвечающий за функции беспроводной связи (wireless communications). И последняя — микропроцессор, выполняющий software-приложения, управляющий всем устройством в целом — операционная система реального времени. Такая структура устройства связана с тем, что реконфигурируемая hardware на ПЛИС не в состоянии самостоятельно обеспечить оптимальное решение всех ситуационных проблем. Поэтому устройство кроме FPGA должно содержать микропроцессор, ASIC и память. При этом имеется возможность маневра средствами при планировании решения задачи на микропроцессоре или в аппаратуре на FPGA, что обеспечивается специальной операционной системой реального времени. Основой проекта является карманный персональный компьютер Compaq iPAD, к которому подключена отладочная плата с двумя микросхемами Xilinx VirtexII FPGA, работающими на частоте 30 МГц. Первая, размером 6 миллионов вентиляей, используется в качестве аппаратного акселератора для игр и видеodeкодирования. Вторая, размером 3 миллиона логических вентиляей, никогда не реконфигурируется, реализует только вложенные в нее вспомогательные функции управления и настройки протоколов конфигурации первой FPGA. Отладочная плата подключается через шину расширения к iPAQ, который содержит процессор StrongARM SA-1110, работающий на частоте 200 МГц, ОЗУ размером 64 М байт и флэш-память 32 М байт. Если представить, что пользователь смотрит фильм со скоростью 25 кадров в секунду и максимальным разрешением 320 x240 пикселей, то для выполнения этой задачи необходимо использовать FPGA. Затем он пожелает запустить игру, оставив вывод на экран видеоизображения с меньшим размером, разрешением и скоростью смены кадров. Теперь игровое приложение будет выполняться на FPGA, а видеоприложение переместится в микропроцессор, поскольку для выполнения этой задачи мощности микропроцессора будет вполне достаточно.

Большинство исследований по динамической реконфигурации вычислительных устройств фокусирует основное внимание на достижении высокой частоты автоматической перестройки системы [14-18]. Однако в *Gecko* проблема реконфигурации рассматривается с совершенно другой точки зрения. В данном устройстве задачи перемещаются, а следовательно, возникает потребность реконфигурации лишь в том случае, когда пользователь решает запустить другое приложение. Поскольку частота вмешательства пользователя, как правило, невелика — секунды, минуты и даже часы, — то время, необходимое на реконфигурацию системы (1-10 мс), будет несравненно меньше длительности выполнения в ней задачи. Если пользователь будет реконфигурировать систему каждые 5 минут, дополнительное время реконфигурации 10 мс будет незаметно. При этом используется частичная и плавная реконфигурация: в момент загрузки в FPGA новой задачи расположенные в остальной части микросхемы приложения продолжают выполняться.

Каждое приложение представляет собой набор модулей для решения конкретных задач. Некоторые из них не требуют большого объема вычислений и могут быть выполнены программным способом на микропроцессоре. Другие, наоборот, для обеспечения высокой скорости выполнения требуют реализации на FPGA. Для того чтобы иметь возможность ускорения нескольких приложений одновременно, микросхема разбивается на три зоны. Тогда количество одновременно выполняемых задач не может превысить числа три. Иначе, ресурсы одной из зон могут быть распределены между несколькими задачами или выполнение некоторых, наименее критичных к быстродействию, будет передано микропроцессору. На практике частичной реконфигурации бывает недостаточно. Поэтому все задачи, выполняемые на FPGA и микропроцессоре, должны иметь гибкие интерфейсные средства коммуникации между собой.

Существенным моментом при создании проекта было обеспечение максимально безконфликтного переключения между приложениями. Эта задача была возложена на операционную систему реального времени *Gecko's Operating System for Reconfigurable System*. Ее функции: управлять сложными переходами и динамическим созданием конфигурации аппаратных средств для решения задач и обеспечения связей между ними; распределять ресурсы для новой задачи, переключать решение задачи между FPGA и микропроцессором, приостанавливать выполнение некоторых задач с запоминанием их состояния в точке останова в целях последующего возобновления решения.

Для того чтобы выполнять безконфликтное и автоматическое переключение задач между FPGA и микропроцессором и наоборот, были рассмотрены традиционные технологии связей микропроцессоров и операционных систем, которые решили половину проблем, относящихся к программной части. В многозадачных ОС каждой из них предо-

ставляется временной интервал. Выполнение задачи X приостановится, если прежде требуется выполнить задачу Y, после чего продолжается решение задачи X. При переключении режимов решения сохраняется вся информация, необходимая для ее восстановления: состояния регистров микропроцессора и памяти, с которым работало приложение, а также счетчика команд.

Однако для реконфигурирования элементов мобильного устройства потребовался специальный подход. Каждая задача имеет аппаратную и программную компоненты для ее реализации, а также специальные фрагменты кода, называемые точками переключения. Когда выполнение задачи доходит до этих точек, задание приостанавливается с последующей передачей информации о состоянии задачи операционной системе, которая сохраняет ее в памяти процессора.

Каждый пользователь компьютера знаком с ситуацией, когда программы автоматически осуществляют поиск в Internet своих обновлений. Таким же образом можно выполнять обновление и аппаратной реализации задач.

Созданием универсальных реконфигурируемых устройств также активно занимаются ученые в лабораториях Калифорнийского университета в Беркли (University of California at Berkeley); Имперского колледжа Науки, Техники и Медицины в Лондоне (Imperial College of Science, Technology and Medicine in London); Массачусетского технологического института (Massachusetts Institute of Technology), где аналогичный *Gecko* проект является частью RAW-процессора (Raw Architecture Workstation).

Gecko — это только первый шаг к созданию гибких реконфигурируемых систем на чипе. В будущем одна микросхема объединит все ранее разработанные компоненты *Gecko*. Новое поколение *Gecko* будет создано при использовании 45 нм технологии, которая появится в 2008-2010 годах (сегодня это пока 90 нм).

5. Универсализация или специализация? Выводы

На основе частичного анализа рынка электронных технологий, касающегося проектирования и производства цифровой и микропроцессорной техники, можно сказать, что каждое из устройств (микропроцессор, PLD, ASIC) является самодостаточным. За неограниченное время на каждом из упомянутых кристаллов можно реализовывать цифровую систему. Но проблема заключается в том, чтобы затем ее можно было выгодно продать. И здесь вступают в силу критерии, оказывающие влияние на конечного потребителя: энергопотребление, быстродействие, полезные свойства, время проектирования, стоимость изделия, реконфигурируемость. Данные критерии не претендуют на полноту, но они существенно влияют на циклические изменения вектора предпочтения рынка в сторону универсализации или специализации цифровых систем. Можно предположить, что каче-

ственно график развития рынка по упомянутым понятиям описывается спиральной моделью, представленной на рис. 1. На каждом следующем цикле или временном интервале, равном восьми годам, рынок имеет более высокий уровень универсализации или специализации по отношению к предыдущему периоду. Следует полагать, что фаза настоящего времени соответствует активизации рынка, направленной в сторону специализации (третья четверть на спиральной модели развития рынка), поскольку фирмы, выпускающие универсальные компьютеры, за последние три года терпят существенные убытки.



Рис 1. Спиральная модель развития рынка

Другой интересный вывод связан с окончательным захватом рынка технологией совместного проектирования аппаратной и программной частей цифровой системы (рис. 2). При этом особенностями современных систем проектирования являются следующие методологии [19-24].

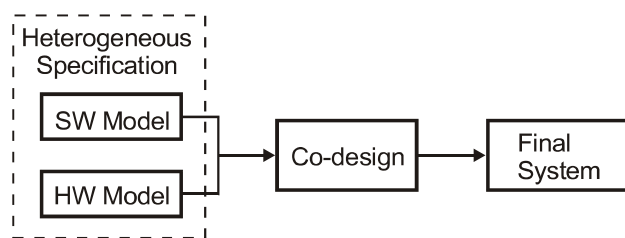


Рис 2. Гетерогенная модель процесса совместного проектирования

– Поддержка нескольких способов формализации спецификаций, что дает возможность: 1) применять метод описания, который является более предпочтительным для адекватного представления функций фрагмента проекта; 2) использовать в качестве компонентов уже верифицированные модели и IP-core, язык описания которых отличается от применяемого в текущем проекте.

– Возможность проведения валидного анализа поведения компонентов в реальном масштабе времени и повторное применение верифицированных проектов (design reuse).

– Наличие нескольких методов формализации спецификаций необходимо для проектирования сложных систем, состоящих из гетерогенных компонентов со смешанными свойствами: аналогово-дискретные, программно-аппаратные, электрон-

но-механические. Это необходимо и для разработки электронной системы несколькими группами проектировщиков, использующими различные методы формализации спецификаций или ранее созданных моделей компонентов, представленных другими языками описания.

– Существующие средства hardware-software co-design системного уровня используют широкий спектр языков описания, среди которых доминируют C++ и/или VHDL.

– Общие для всех ведущих фирм стадии разработки: 1) Создание неформальной модели всей системы в целом. 2) Формализация спецификации проекта. 3) Разработка архитектуры и разделение системы на фрагменты основывается на наличии конкретных средств проектирования и инженерном опыте проектировщика. Окончательный вариант архитектуры системы является результатом удачных и неудачных последовательно выполняемых итераций доработки формальной спецификации. 4) Разработка hardware и software для компонентов, составляющих проект, осуществляется параллельно или независимо.

Литература: 1. Bass M.J. and Cristensen C.M. The future of the Microprocessor Business. IEEE Spectrum, April, 2002, P. 34-39. 2. Philip E. Ross. "5 Commandments". IEEE Spectrum. December 2003. P. 30-35. 3. Nick Tredennock, Brion Shimamoto. "Go Reconfigure". IEEE Spectrum. December 2003. P. 37-40. 4. Benini L., Macii A., Macii E., Omerbegovic E., Poncino M., Pro F. Energy-Aware Design Techniques for Differential Power Analysis Protection // Design Automation Conference (DAC 03). 2003. P. 36-42. 5. Gupta B., Borgatti M. Different Approaches to Add Reconfigurability in a SoC Architecture // Design Automation and Test in Europe (DATE 03). 2003. P. 398-402. 6. Blodget B., McMillan S., Lysaght P. A Lightweight Approach for Embedded Reconfiguration of FPGAs // Design Automation and Test in Europe (DATE 03). 2003. P. 399-401. 7. Walder H., Platzner M. Online Scheduling for Block-Partitioned Reconfigurable Devices // Design Automation and Test in Europe (DATE 03). 2003. P. 290-296. 8. Geppert L. The New Indelible Memories // IEEE Spectrum. March 2003. P. 49-54. 9. M. Borgatti, L. Calm, G. De Sandre, B. Forkt, D. Iezzi, F. Lertora, G. Muzzi, M. Pasotti, M. Poles, P.L. Rolandi (STMicroelectronics) A Reconfigurable Signal Processing IC with Embedded FPGA and Multi-Port Flash Memory // Design Automation Conference (DAC 03). 2003. P. 691-696. 10. Amlani I., Zhang R., Tresek J., Nagahara L., Tsui R. K. Manipulation and Characterization of Molecular Scale Components // Design Automation Conference (DAC 03). 2003. P. 276-278. 11. Don Edinfeld, Andrew B Kahng, Mike Rodgers, and Yervant Zorian. 2003 Technology Roadmap for Semiconductors // Computer, January, 2004. P. 47-56. 12. Diederik Verkest. Machine Chameleon // IEEE Spectrum. December, 2003. P. 41-46. 13. Mei B., De Man H., Lauwereins R., Vernalde S., Verkest D. Exploiting Loop-Level Parallelism on Coarse-Grained Reconfigurable Architectures Using Modulo Scheduling // Design Automation and Test in Europe (DATE 03). 2003. P. 292-306. 14. Vissers K. Parallel Processing Architectures for Reconfigurable Systems // Design Automation and Test in Europe (DATE 03). 2003. P. 396-398. 15. Spirakis G. Leading-Edge and Future Design Challenges-Is the Classical EDA Ready // Design Automation Conference (DAC 03). 2003. P. 416-418. 16. Choi K., Kim K., Pedram M. Energy-Aware MPEG-4 FGS

Streaming // Design Automation Conference (DAC 03). 2003. P. 912-916. **17.** *Mooney V.* III Hardware/Software Partitioning of Operating Systems // Design Automation and Test in Europe (DATE 03). 2003. P. 338-340. **18.** *Lange S., Kebschull U.* Virtual Hardware Byte Code as a Design Platform for Reconfigurable Embedded Systems // Design Automation and Test in Europe (DATE 03). 2003. P. 302-309. **19.** *Yoo S., Jerraya A.* Introduction to Hardware Abstraction Layers for SoC // Design Automation and Test in Europe (DATE 03). 2003. P. 336-338. **20.** *Nikolaos S. Voros, Sanchez Luis, Alonso Alejandro, Alexios N. Birbas, Michael Birbas, Ahmed Jerraya.* Hardware-Software Co-Design of Complex Embedded Systems. Design Automation for Embedded Systems. Boston: Kluwer Academic Publishers. 2003. P. 5-34. **21.** *Boehm B.* A Spiral model of Software Development and Enhancement. Science of

Computer Programming. 1988. Vol. 8. P. 231-274. **22.** *Vercauteren, S., J.V.D. Steen, and D Verkest.* Combining Software Synthesis and Hardware-Software Interface Generation to Meet Hard Real-Time Constraints. DATE'99. 1999. P. 177-187. **23.** *Hard, D.* StateCharts. A Visual Formalism for Complex Systems. 1987. Vol. 21. P. 77-82. **24.** *Jerraya A.A.* System Level Synthesis SLS. TIMA Laboratory. Annual Report 2002. P. 65-75.

Поступила в редколлегию 11.02.2004

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Хаханова Ирина Витальевна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и диагностика цифровых устройств. Увлечения: английский язык, гитара. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua

УДК 681.325

ВЕРОЯТНОСТНЫЙ ПОДХОД К ФУНКЦИОНАЛЬНОМУ ДИАГНОСТИРОВАНИЮ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ ДЛЯ ОБРАБОТКИ ПРИБЛИЖЕННЫХ ДАННЫХ

ДРОЗДА В.

Рассматривается вопрос функционального диагностирования вычислительных устройств, выполняющих обработку приближенных данных. Предлагается вероятностный подход к диагностированию устройств, повышающий достоверность контроля приближенных результатов путем снижения отбраковки достоверных результатов. Приводятся оценки снижения показателя недостоверности контроля результатов при обработке приближенных данных.

1. Введение

Различают тестовое и функциональное (рабочее) диагностирование цифровых устройств. Оба вида диагностирования определены для решения задачи оценки технического состояния устройства, достоверности его функционирования: тестовое — в паузах основной работы устройства, а функциональное — в процессе его использования по назначению, на рабочих воздействиях [1]. Однако функциональное диагностирование решает еще одну задачу — оценку достоверности получаемых результатов. Значимость этой задачи следует из утвердившегося в функциональном диагностировании требования по обнаружению в самопроверяемых схемах неисправности по ее первому проявлению в виде ошибки [2]. Известно, что ошибки значительно чаще вызываются сбоями — кратковременными самоустраняющимися неисправностями, нежели отказами, дающими устойчивое нарушение работоспособности схем [3]. Сбой в дальнейшем не проявляется и не влияет на техническое состояние устройства, однако приводит к искажению результата. Поэтому функциональное диагностирование в первую очередь направлено на оценку достовер-

ности результатов. Однако данная задача, как правило, отождествляется с оценкой достоверности функционирования устройства, что можно объяснить принятием модели точных вычислений. Эта модель состоит в том, что контроль вычислительных операций выполняется без учета природы чисел, однозначно относя их к точным данным. Благодаря этой модели в функциональном диагностировании предъявляются единые требования к управляющим и вычислительным схемам, поскольку любая ошибка в управляющем слове так же недопустима, как и в точном данном. На основе модели точных вычислений устанавливается взаимно-однозначное соответствие между достоверностью функционирования устройства и достоверностью его результатов, что позволяет подменить цель — получение достоверных результатов на средство ее достижения — обеспечение соответствующего технического состояния устройства.

Однако модель точных вычислений сложилась в условиях ограниченных стартовых возможностей вычислительной техники и в настоящее время перестает адекватно отражать действительность. Основную часть чисел составляют приближенные данные. К точным данным относятся лишь числа, являющиеся целыми по своей природе — номера элементов множеств. Остальные данные — результаты измерений и их обработки — являются приближенными.

Приближенный характер данных существенно меняет условия функционального диагностирования вычислительных устройств, требуя разработки новых подходов для оценки достоверности результатов.

2. Постановка задачи

В вычислительной математике различаются старшие верные разряды приближенного числа и младшие неверные разряды, находящиеся по весовым функциям ниже уровня абсолютной погрешности [4]. Обработка приближенных данных выполняется с отбрасыванием заведомо (согласно теории ошибок [5]) неверных разрядов. Ошибки, вызванные неисправностью схемы в отбрасываемых и оставляемых неверных разрядах, не влияют на достоверность результатов приближенных вычис-